UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Construction of a Radio Environment Map of TV White Spaces for Barcelona Metropolitan Area

A Degree's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Martí Cervià Caballé

In partial fulfilment

of the requirements for the degree in

Science and telecommunication Technologies
ENGINEERING

Advisor: Ferran Casadevall

Barcelona, February 2015

Abstract

This thesis covers the construction of a REM (Radio Environment Map) for TV-WS (TV white spaces) for the metropolitan area of Barcelona, and a GUI for accessing it.

A REM is a key piece in the implementation of a cognitive network since proportions environmental information to cognitive devices, allowing them to decide what channels they have to use, and when they have to use them in order to perform optimally.

The term TV-WS refers to the spectral zones that are assigned to TV broadcasting, but they are not used for that purpose since the transition to the DVB-T standard. This new standard allowed for better spectral efficiency in TV broadcasting, fact that let to the underutilization of the TV assigned spectrum. This underutilized spectrum could be used opportunistically for unlicensed users (cognitive users) to establish a REM backed cognitive network.

Field measurements from the metropolitan area of Barcelona are used for REM construction, which had been posteriorly processed to adapt them to the REM structure. The access to the REM is carried over a REST API, which is also used to administrate it. The GUI has been implemented as a webpage that can be used to access the REM data, and to administrate the REM. Finally an interface which permits to use the REM to manage simulated cognitive devices has been developed, in order to showcase the capabilities of this technology.

Resum

Aquest treball detalla la construcció d'un REM (Radio Enviroment Map) de TV-WS (TV white spaces) a partir de mesures de canal preses a l'entorn de l'àrea metropolitana de Barcelona, i de una GUI per permetre la seva consulta.

Un REM es una peça clau en la implementació d'una xarxa cognitiva, ja que proporciona als dispositius cognitius coneixement del medi radio en el que es troben, permetent als dispositius decidir quins canals han de fer servir, i quan els han d'utilitzar per operar de forma òptima.

Els TV-WS son les zones espectrals que estan assignades a televisió, però no s'usen per a aquest servei des de la transició a l'estàndard DBTV, que va augmentar considerablement la eficiència espectral de les emissions de televisió deixant gran part de l'espectre assignat, sense utilitzar. Aquest espectre es podria utilitzar de forma oportunista per a usuaris sense llicencia (usuaris cognitius) per a establir comunicacions entre ells, mitjançant una xarxa cognitiva basada en REM.

Per a la construcció del REM s'han utilitzat mesures de camp preses a l'àrea metropolitana de Barcelona, que han estat processades posteriorment per adaptar-se a l'estructura REM. La informació continguda al REM es pot consultar a través de una API REST, que a més a més es pot utilitzar per administrar el REM. La interfície de consulta del REM finalment ha estat construïda mitjançant una pàgina web que permet visualitzar i administrar el REM. A més a més s'ha creat una interfície que permet utilitzar el REM per administrar dispositius cognitius simulats amb l'objectiu de demostrar les capacitats de la tecnologia.

Resumen

Esta tesis describe la construcción de un REM (Radio Enviroment Map) de TV-WS(TV White Spaces) usando medidas de canal tomadas en la área metropolitana de Barcelona, además de una GUI para facilitar su consulta.

Un REM es una pieza clave para la implementación de una red cognitiva, ya que proporciona a los dispositivos información del medio radio en el que se encuentran, permitiéndoles que canales utilizar, y cuando utilizaros para operar de forma óptima.

Los TV-WS son las zonas espectrales asignadas a televisión pero que no se usan para ese cometido desde la transición al estándar DVB-T, que aumentó en gran medida la eficiencia espectral de estas emisiones, dejando gran parte del espectro asignado a usos televisivos sin utilizar. Este espectro podría utilizarse de forma oportunista por parte de usuarios sin licencia (usuarios cognitivos) para establecer conexiones entre ellos, a través de una red cognitiva basada en REM.

Para la construcción del REM se han usado muestras de campo tomadas en el área metropolitana de Barcelona, que han sido posteriormente procesadas con el objetivo de adaptarlas a la estructura REM. Se puede acceder a la información contenida en el REM a través de una API REST, que, además, permite administrarlo. La GUI ha sido implementada a través de una página web, que permite visualizar y administrar el REM. Finalmente, se ha añadido una interfaz que permite utilizar el REM para administrar dispositivos cognitivos simulados, con el objetivo de demostrar las capacidades de esta tecnología.

Per la Laura. Sense tu, res tindria sentit.

Acknowledgements

Firstly, I want to thank my supervisor, Ferran Casadevall for his thoughtful advice and the editorial assistance in the development of this thesis. I also want to thank Sergio Garcia for his advice and helping me put my ideas in order for the web development part of the project.

Revision history and approval record

Revision	Date	Purpose
0	19/1/2015	Document creation
1	18/1/2015	Document revision
2	2/1/2015	Document revision
3	5/1/2015	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Martí Cervià Caballé	marti.cervia@alu.upc.edu
Ferran Casadevall	ferranc@tsc.upc.edu

Written by:		Reviewed and approved by:	
Date	19/1/2015	Date	5/1/2015
Name	Martí Cervià	Name	Ferran Casadevall
Position	Project Author	Position	Project Supervisor

Table of contents

The table of contents must be detailed. Each chapter and main section in the thesis must be listed in the "Table of Contents" and each must be given a page number for the location of a particular text.

Abstract 1
Resum 2
Resumen 3
Acknowledgements 5
Revision history and approval record 6
Table of contents 7
List of Figures 9
List of Tables: 10
1. Introduction: 11
1.1. Purpose: 11
1.2. Background: 11
1.3. Requirements and Objectives: 13
1.4. Previous works: 13
1.5. Work plan: 15
1.6. Gantt Diagram: 19
1.7. Deviations and incidences: 20
2. State of the art of the technology used or applied in this thesis: 21
2.1. Cognitive Radio, an introduction: 21
2.2. Cognitive Radio, use cases: 22
2.3. TV-WS: 23
2.4. REM based Cognitive Radio: 23
2.5. REM in TV-WS: 24
2.6. Current REM enabled CR standards, IEEE 802.22 WRAN: 24
3. Methodology / project development: 25
4. Results: 37
5. Budget: 44
5.1. Labor costs: 44
5.2. Software costs: 44
5.3. Amortizations: 44

6. Conclusions and future development: 45
6.1. Summary: 45
6.2. Future Development: 45
Bibliography: 46
Appendices: 48
GUI User's manual: 48
API documentation : 51
Interpolation methodology: 53
Glossary: 54

List of Figures

Figure 1: Gantt Diagram……19
Figure 2: Simplified version of the cognitive cycle…… 21
Figure 2: Duty cycle statistics…….. 22
Figure 4: Relation between the spectrum manager and other cognitive entities…. 24
Figure 5: REM architecture……... 25
Figure 6: sample GUI screenshot…… 37
Figure 7: sample GUI heat map visualization…… 38
Figure 8: devices operating outside the selected channel……... 38
Figure 9: Device 2 in the original 500m grid……... 39
Figure 10: Device 2 in the updated 1km grid…….. 40
Figure 11: app interface…… 41
Figure 12: app results…… 42
Figure 13: results of the synthetic benchmark…… 42
Figure 14: Alarm Probability versus sample size…….. 43
Figure 15: Primary Not Detected versus sample size…… 43
Figure 16: Elements of the GUI…… 48
Figure 17: channel selection…….. 49
Figure 18: grid selection…… 49
Figure 20: grid style selection…… 50
Figure 21: display selection…… 50
Figure 22: REM parameters……... 50
Figure 23: delete devices…… 50
Figure 24: insert devices…… 50

List of Tables:

Table 1: Work package 1..... 15
Table 2: Work package 2..... 15
Table 3: Work package 3..... 16
Table 4: Work package 4..... 16
Table 5: Work package 5..... 17
Table 6: Work package 6..... 18
Table 7: Work package 7..... 18
Table 8: Labour costs..... 44
Table 9: Software costs..... 44
Table 10: Depreciation..... 44

1. Introduction:

1.1. Purpose:

The transition from analog TV to digital TV has left a great deal of the UHF spectrum vacant. However, this part of the spectrum is still allocated for broadcasting purposes. In addition, an added difficulty for using this free band is that the vacant channels are not the same at all locations, even for relatively small geographical areas. With the aim of facilitating the use of the free spectrum without creating interference for the licenced users, we propose the construction of a REM for TV-WS as the first step for the creation of a test bed for cognitive network.

A REM is a system that maintains an updated database of the behaviour of the radio medium at a given geographical area. This system can be used to provide information about the usage of the radio spectrum at a selected area, enabling connected devices to know when a certain channel it's not used by it's licensed users in order access it. This type of access constitutes opportunistic usage of the spectrum, and it is one of the main applications of cognitive radio technology.

This thesis covers the design and implementation of a REM manager, that will construct the REM from previously taken field samples, and a GUI component that facilitates using the REM.

1.2. Background:

Nowadays one of the main problems that the telecommunications industry has to face is that almost all the spectrum useful for wireless radio communications is already allocated. However, the amount of wireless data bandwidth needed is always increasing. This situation poses a very serious technical problem if the current growth of mobile communications has to be maintained.

As it is mentioned before, the majority of the useful spectrum is already allocated; this means that is reserved for a certain use and that has a specific licensed user. Only the licensed users, and no one else are allowed to make use of their allocated spectrum because, if someone else makes use of it, the licensed user could be interfered.

However, that a certain part of the spectrum is allocated does not mean that it's owner is using it all the time, or in every geographic location. The lack of usage of certain radio channels means that there is room for improvement of the usage of the channel. One

could think that a third party, or secondary user, could make use of allocated spectral bands when or where their rightful owner, or primary user, is not making use of them. This constitutes the opportunistic usage of the spectrum. Opportunistic usage of spectrum is one of the tools that will allow the capacity of wireless networks to grow and face the expected increase of wireless traffic, as it maximizes the usage of the available spectrum. The technology that allows for the dynamic usage of spectrum to happen is Cognitive Radio, which is listed as one of the technologies that will allow 5G wireless communications to be developed. A cognitive radio is a terminal that is aware of its immediate environment and can change dynamically its transmission or reception characteristics in order to be adapt to it. This adaptability allows it to hop between frequency bands if necessary, a feature needed for the opportunistic usage of the spectrum, as it allows for a secondary user to clean the channel if it senses that a primary user is transmitting.

The mechanism that allows a cognitive terminal to be aware of its surroundings and detect if it's interfering a primary user is called spectrum awareness. It can be implemented in different ways. One is having a geo-localization device in the terminal, and use the obtained location to consult an external agent what spectrum zones are available for use at the current location. In this technique, the devices consult to a REM or Radio Environment Map. A REM is a system that, combines up to date information about the radio characteristics of the environment (measurements taken by the terminals, local policies, known positions or usage patterns of the primary and secondary users etc.) and uses them to maintain a real time map of the radio scene of a certain area and to transmit to the cognitive terminals where they can transmit and in what frequencies.

A REM allows for an easy and economical implementation of a cognitive network, because it allows the cognitive terminals to be relatively simple since the REM takes the most complex part of the spectrum awareness task. Nowadays the right conditions have appeared to begin to implement REM based cognitive networks.

Even though, the concept of cognitive radio and, in particular, the use of a REM for providing Spectrum Awareness to the cognitive devices is something general, the TV band, ranging from 470MHz to 790 MHz in Spain, is specially useful for implementing the cognitive radio concept.

The opportunity for implementing cognitive radio technology lies in the transition between analog and Terrestrial Digital Television. Digital channels occupy 4 times less band that analog ones, and also, the need to keep certain channels empty to avoid interference between TV repeaters is gone, since Terrestrial Digital TV is a synchronous network. These factors have vacated most of the space between 470-790 MHz; the unused channels constitute TV White Spaces (TV-WS). This white spaces would be useful in a series of wireless applications and services, such as Wireless Sensor Networks (WSN), indoor WLANs, or rural broadband access, all of which can be deployed under the concept of cognitive radio. The main challenge for TV-WS utilization is that they are not geographically uniform, even within a certain region. This fact is due that some repeaters transmit local channels, and the coverage of these repeaters varies.

In summary, the usage of TV white spaces would be very beneficial from the perspective of technological and economical development of these new services, but requires the implementation of new techniques such as cognitive radio. The best solution from the technical perspective would be a cognitive network based in REM architecture. A REM implementation would be optimal for the TV-WS situation since, the radio scene is mostly

static, and if a channel is vacant is highly probable that it will remain vacant for a great period of time. This allows for a simple REM implementation, since the primary user activity is mostly static and predictable and the refresh ratio of the REM information is very low. Finally, it is not worthless to mention that the implementation of a REM will provide the GRCM research group with a tool for accessing the information about the radio scene easily and in real time as well as a key component to implement a test bed for cognitive devices.

1.3. Requirements and Objectives:

This project corresponds to an early prototype/ feasibility test. Even thought this project does not have many concrete requirements since it does not have to adhere to a standard or implement a market product, still, there are two main requirements. One is to complete a valid implementation of a REM system with a functional GUI at the end of the project and the other is to validate the interpolation algorithm used in the construction of the REM manager, a key functionality in the REM implementation.

From the requirements the following set of goals can be devised:

1. Propose a suitable REM architecture.
2. Select and validate an interpolation algorithm to be used in the REM manager in order to approximate the REM data at areas not covered by the measurements.
3. Create a GUI tool to easily access the data stored in the REM.
4. Integrate all the REM components into a single system.

1.4. Previous works:

This project uses the measurements taken in the following projects:

• Pedro Delicado Alcántara, “medidas para la construcción de un REM (Radio Environment Map) de TV-WS en la mitad norte del Ária Metropolitana de Barcelona”, 17/9/2014

• Gustavo Herraiz Montes, “medidas para la construcción de un REM (Radio Environment Map) de TV-WS en la mitad sur del Ária Metropolitana de Barcelona”, to be presented.

For the development of this project the following 3rd party software tools have been considered:

- Matlab: for the data processing and interpolation.
- GDAL: for converting the interpolated data to GeoJsons for the REM.
- node.js: for the development of the back-end REM software.

And the following libraries and frameworks:

- geojson-utils: a set of utilities for working with geojson objects in javascript.
- Express.js : a flexible web application development framwork.
- Node-sqlite3: a library that provides bindings for using an sqlite database from node.js.
- Google Maps Javascript API v3: a framework to manipulate and create maps for browsers.

All the software components, except Matlab, are open source, geojson-utils is distributed under the BSD license and the rest of open source components are distributed under the MIT license.

This work has been developed in the framework of the research project “Redes móviles eficientes para la ampliación de servicios a nuevos sectores profesionales (RAMSES)”, founded by Comisión Interministerial de Ciencia y Tecnología (C.I.C.Y.T.), project referenceTEC2013-41698-R

1.5. Work plan:

<table>
<tr><td colspan="2">Project: Construction of a Radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area</td><td>WP ref: 1</td></tr>
<tr><td colspan="2">Major constituent: Planning and Research</td><td>Sheet 1 of 7</td></tr>
<tr><td colspan="2" rowspan="2">Short description:
Do an initial bibliographical research about the topic in the literature and define the project scope, goals, timeplan, work packages and interaction procedures with the advisor</td><td>Real start date: 15/9/2014
Real end date: 10/10/2014</td></tr>
<tr><td>Start event:
End event:</td></tr>
<tr><td>Internal task T1: Bibliographic research

Internal task T2: Project Planning</td><td>Deliverables:
Project Proposal and Workplan</td><td>Dates:
9/10/2014</td></tr>
</table>

Table 1: Work package 1

<table>
<tr><td colspan="2">Project: Construction of a Radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area</td><td>WP ref: 2</td></tr>
<tr><td colspan="2">Major constituent: REM architecture definition (design)</td><td>Sheet 2 of 7</td></tr>
<tr><td colspan="2" rowspan="2">Short description:

Define the components of the REM implementation, the nodes involved and the interactions between them.</td><td>Planned start date: 6/10/2014
Real start date: 9/10/2014

Planned end date: 15/10/2014
Real end date: 15/10/2014</td></tr>
<tr><td>Start event:
End event:</td></tr>
<tr><td>Internal task T1: Definition of the nodes involved.

Internal task T2: Definition of the interconnections between nodes.</td><td>Deliverables:
REM architecture interim working document</td><td>Dates:
15/10/2014</td></tr>
</table>

Table 2: Work package 2

<table>
<tr><td colspan="2">Project: Construction of a Radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area</td><td>WP ref: 3</td></tr>
<tr><td colspan="2">Major constituent: Interpolation Methodology(Research and software)</td><td>Sheet 3 of 7</td></tr>
<tr><td colspan="2" rowspan="2">Short description:
Review and comparison selection and implementation of a interpolation method.</td><td>Planned start date: 10/10/2014
Real start date: 12/10/2014

Planned end date: 31/10/2014
Real end date: 3/11/2014</td></tr>
<tr><td>Start event:
End event:</td></tr>
<tr><td>Internal task T1: Review and comparison of the various methods.

Internal task T2: Selection of the method to implement.

Internal task T3: Software implementation.

Internal task T2: Validation of the GUI software</td><td>Deliverables:
Implemented interpolation method program</td><td>Dates:
3/11/2014</td></tr>
</table>

Table 3: Work package 3

Project: Construction of a Radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area	WP ref: 4
Major constituent: Creating a GUI (software)	Sheet 4 of 7
Short description: Elaborate the requirements for the GUI program and select an adequate technology for the implementation. Then implement it and validate the software produced.	Planned start date: 20/10/2014 Real start date: 27/10/2014 Planned end date: 19/11/2014 Real end date: 1/12/2014

	Start event: End event:	
Internal task T1: Requirements. Internal task T2: Technology assessment. Internal task T3: Implementation Internal task T2: Validation	Deliverables: GUI software	Dates: 1/12/2014

Table 4: Work package 4

Project: Construction of a Radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area	WP ref: 5	
Major constituent: Integration and validation (software)	Sheet 5 of 7	
Short description: Integrate all the nodes of the REM and validate its correct operation.	Planned start date: 20/11/2014 Real start date: 3/11/2014 Planned end date: 19/12/2014 Real end date: 18/1/2015	
	Start event: End event:	
Internal task T1: Integration of the nodes and interconnections. Internal task T2: Validation of the REM.	Deliverables: Integrated REM prototype	Dates: 19/12/2015

Table 5: Work package 5

Project: Construction of a radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area	WP ref: 6	
Major constituent: Create a mobile Demo application (Extra-work not planned initially)	Sheet 6 of 7	
Short description: Create a demo mobile application that will be used to show the REM working.	Planned start date: 22/12/2014 Real Start Day : 20/1/2015 Planned end date: 9/1/2015 Real end day: 25/1/2015	
	Start event: End event:	
	Deliverables: Demo mobile application	Dates: 30/1/2015

Table 6: Work package 6

Project: Construction of a radio Environment Map (REM) of TV White Spaces for Barcelona Metropolitan Area	WP ref: 7	
Major constituent: Final Report (writing)	Sheet 7 of 7	
Short description: Develop the final report of this project using an iterative methodology.	Planned start date: 11/12/2014 Real start date: 19/1/2015 Planned end date: 30/1/2015 Real end date: 6/2/2015	
	Start event: End event:	
Internal task T1: Writing the final report Internal task T2: Revising the final report	Deliverables: Final Report	Dates: 30/1/2015

Table 7: Work package 7

1.6. Gantt Diagram:

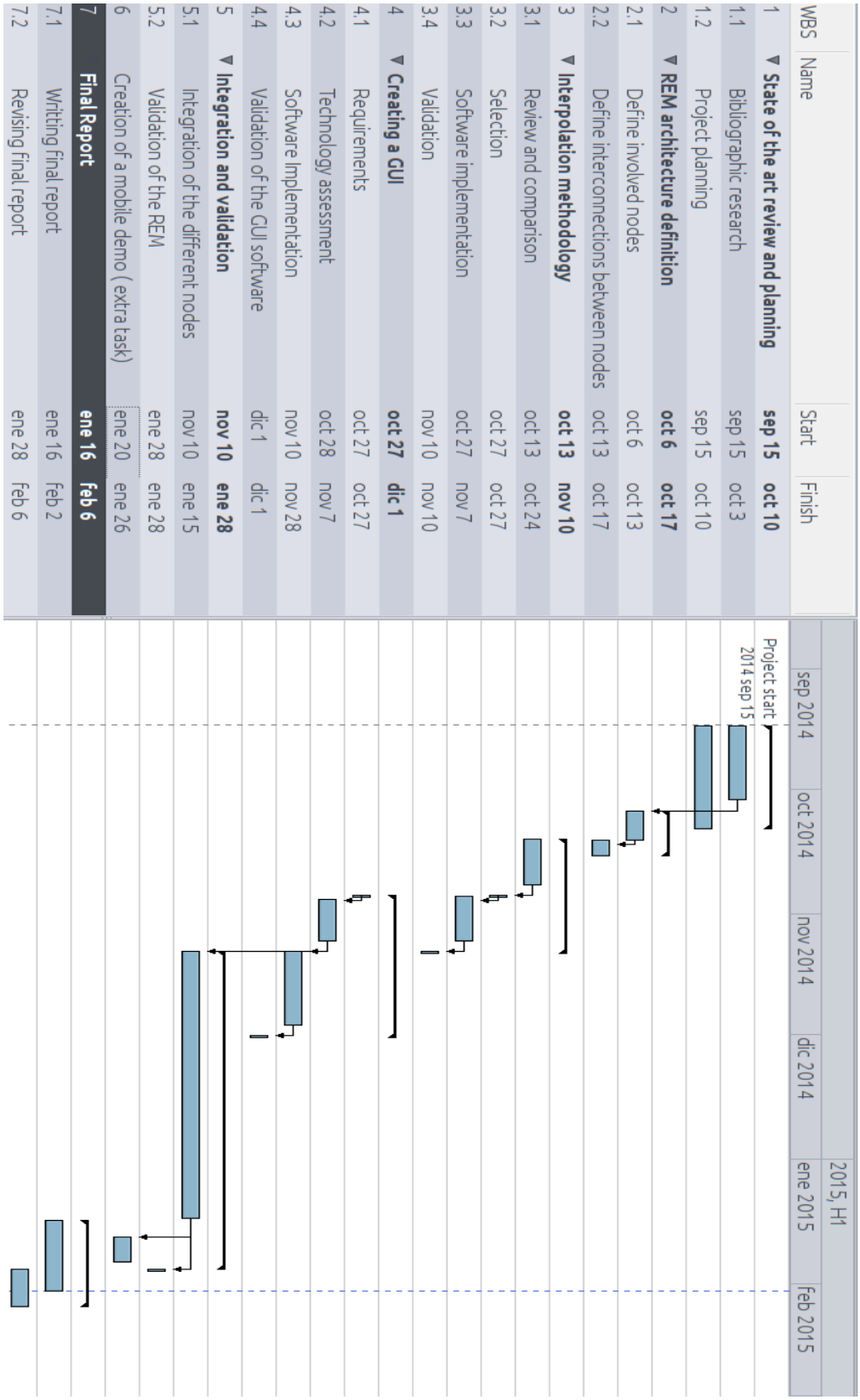


Figure 1: Gantt Diagram

1.7. Deviations and incidences:

The first design contemplated creating the REM as a web application, using a Google service called Google Maps Engine as back-end provider for the REM. Latter this year, Google announced that it would close the service soon, so a new and dedicated back-end program had to be created for the REM. This unexpected development increased the length of the integration tasks, and also caused delays the development of the GUI, since parts of the back-end had to be integrated before making a final version of it, and also to adapt parts of the Matlab script used to build the REM data.

The former circumstances pushed back the delivery of the GUI to the start of December, and also lengthened the integration tasks to mid January.

Moreover, an extension task was added to the initially devised tasks. This new task was to create a simple android app to showcase the REM API functionality. The app used the integrated geo-localtion capabilities of the smart-phone to insert a simulated device at the REM. The app sent a request to the REM manager with the position of the device and the required bandwidth, and then received a response from the REM manager with the simulated device characteristics.

2. State of the art of the technology used or applied in this thesis:

2.1. Cognitive Radio, an introduction:

The key technology of this thesis is cognitive radio. The term cognitive radio was first used in an article “Cognitive Radio: An integrated agent architecture for software radio architecture” [1], and is defined as a radio that is “self aware” in terms of its operational characteristics. It extracts information of its physical environment and uses it to adapt to the current conditions, and can exchange and interpret radio environment information with other entities with the objective of providing reliable communications when needed, and using efficiently the available spectrum [2].

The adaptability of cognitive radio derives from the concept of software radio and software defined radio. Software Radio is a radio where all the RF processing, including RF channel access, is defined on a general computing platform using software, commonly using high performance DSPs or FPGAs to process it. Software defined radio is an architecture where not all the processing is necessarily implemented by software, but at least, is configurable by software by switching between different ASICs and other hardware modules [2]. The fact that all processing is configurable by software provides an incredible grade of flexibility to the system, enabling it to switch between different parts of the spectrum, and using different modulations and wireless standards.

Cognitive radio ads a layer of environmental awareness to a software radio or software defined radio architecture, enabling it to change it's transmission or reception parameters autonomously in order to adapt to the current state of the local radio channel. The process of sampling the channel, reviewing the acquired information and taking a decision based on the obtained and prior knowledge is called the cognitive cycle. A simplified version of the cognitive cycle is represented in [figure 2].

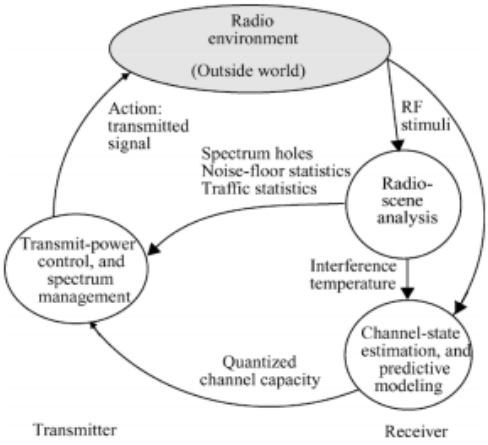


Figure 2: Simplified version of the cognitive cycle. Haykin, Simon, "Cognitive radio: brain-empowered wireless communications," Selected Areas in Communications, IEEE Journal on , vol.23, no.2, pp.201,220, Feb. 2005

2.2. Cognitive Radio, use cases:

One of the challenges that modern wireless communications have to face is the lack of unassigned spectrum and the lack of utilization of some spectral bands. The present regulatory framework for the usage of the electromagnetic radio spectrum is very restrictive, enforcing that only the assigned users can operate in a certain spectral band. The effect of this strict policies is that some parts of the spectrum are not utilized (spectral blanks) continuously or that, in some geographical locations are completely non used [4]. A more local example, can be found in [5], where we can find that the average duty cycle for the zone between 75MHz and 3GHz is only 22.57 % [Figure 3]. Therefore, there is much room for improvement in the utilization of the radio spectrum.

TABLE II. AVERAGE DUTY CYCLE STATISTICS.

Block	Frequency range (MHz)	Average duty cycle		
1	75 – 500	60.98 %	58.65 %	22.57 %
2	500 – 1000	56.67 %		
3	1000 – 1500	2.07 %	5.89 %	
4	1500 – 2000	12.08 %		
5	2000 – 2500	7.57 %		
6	2500 – 3000	1.85 %		

Figure 3: Duty cycle statistics. Casadevall, F.; Umbert, A., "REM-based real time testbed: A proof of concept on the benefits of using REM for improving radio resource management capabilities," Telecommunications (ICT), 2013 20th International Conference on , vol., no., pp.1,5, 6-8 May 2013

Cognitive radio has been proposed for optimizing the usage of the spectral blanks [1],[3] do to it's awareness features and adaptability, that enable a cognitive device to detect when a certain part of the spectrum is not occupied, and change its transmission or reception characteristics to make use of this unused channel, or spectrum blank, becoming this device a Secondary User for that spectrum. Moreover, a cognitive device would be capable of detecting if a licenced user, or primary user, is using the channel, and move its activity to another spectral blank, minimizing the interference endured by primary users.

2.3. TV-WS:

One case of underutilized spectrum is the case of TV white spaces (TV-WS). TV-WS are the outcome of the transition from analogical TV systems to the more spectral efficient DVB-T standard. A study of the benefits /advantages in terms of spectrum usage for Germany and other European countries can be found at [6]. The result of the transition is that certain parts of the allocated TV spectrum was left unused, with variable geographical availability due to different national and local TV broadcasting scenarios.

The TV-WS spectra in Europe is still allocated for TV use, and a regulatory framework is on the way for its utilization with Cognitive radio technology [7], via spectrum sharing contracts between the spectrum holder and another party interested in using the spectrum.

The prospective uses for the TV-WS include indoor LAN systems, rural broadband and wireless sensor networks [8].

2.4. REM based Cognitive Radio:

A Radio Environment Map (REM) characterizes the radio medium at a certain geographical area. It contains information on the entire radio scene: the Primary Users, the Secondary Users, the terrain, the regulatory framework [9].

A cognitive device can perform the spectrum awareness task of the cognitive cycle by using a REM to obtain information about the radio medium, given its location. Using a REM has an inherent advantages against using the device own sensors for detecting the primary user such as hidden node detection problem [10], and can reduce the cost of the Cognitive Devices since their spectrum sensing capabilities can be reduced, or eliminated if the REM uses an independent sensor network.

The main drawback of REM based cognitive radio network is that it needs a dedicated channel for communication between the REM, the cognitive devices and/or the sensor network. This limitation can be somehow limited by using the ISM band as signalling channel [11].

Another drawback for a REM based approach is how dynamic is the radio medium. If the medium changes fast, REM information will become out-dated rapidly. This causes overhead in the network as synchronization for the new data is required, or capacity issues in the designated channel for sharing REM information, as explained in [12].

2.5. REM in TV-WS:

The characteristics of REM enabled cognitive radio are suitable for TV-WS. The spectrum activity at the TV band is almost static, such as the locations of the primary users, as described in [8]. However, a static database is not enough since the Secondary User (SU) activity is dynamic and knowledge of SU activity is needed for cognitive radio. In addition a REM based approach is also useful for keeping the cost of cognitive devices down, a needed feature for CR applications in TV-WS as described in [8].

2.6. Current REM enabled CR standards, IEEE 802.22 WRAN:

The first Cognitive Radio standard is the IEEE 802.22 [13] aimed at providing broadband wireless access at rural areas, using the TV-WS spectrum due to its excellent propagation characteristics.

The standard defines 3 types of devices. An incumbent database, that acts as a global REM, base stations (BS) where each one that creates its own wireless cell, acts as local REM and as coordinator between the different devices in the cell, and Customer Premises Equipment (CPE) that act as sensors for the REM, and end nodes for the cognitive network.

[Figure 4] shows the relations between the three entities mentioned before.

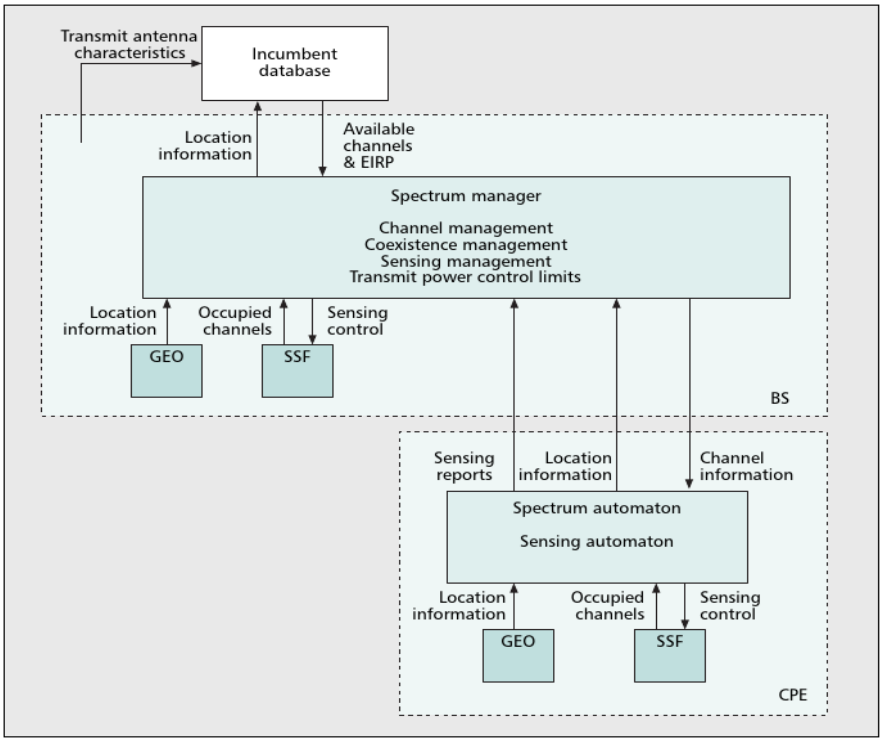


Figure 4: Relation between the spectrum manager and other cognitive function entities. Stevenson, C.; Chouinard, G.; Zhongding Lei; Wendong Hu; Shellhammer, S.J.; Caldwell, W., "IEEE 802.22: The first cognitive radio wireless regional area network standard," Communications Magazine, IEEE , vol.47, no.1, pp.130,138, January 2009

3. Methodology / project development:

1. Rem Architecture:

The first aspect of the construction was to decide on architecture for the implementation of the REM. In order to implement a functional REM, and a GUI to comply with the project objectives, three functional blocks were implemented:

- The REM manager, which is tasked with processing the available samples and perform interpolation in order to construct the REM and managing the cognitive users.
- A storage database, to store the REM data and device list.
- And a GUI to visualize the REM data.

This architecture, excluding the interpolation tasks, is implementable as a web application, being the GUI a web page, and using a back-end that acts as REM manager, responding to cognitive users requests.

As the interpolation tasks involved in REM construction, was not manageable for a standard web application back-end, this functionality had to be segregated to a subdivision of the REM manager block, implemented in Matlab, that performed the interpolation of the primary user data and provided the results to the back-end of the web application.

The final block diagram, was the following:

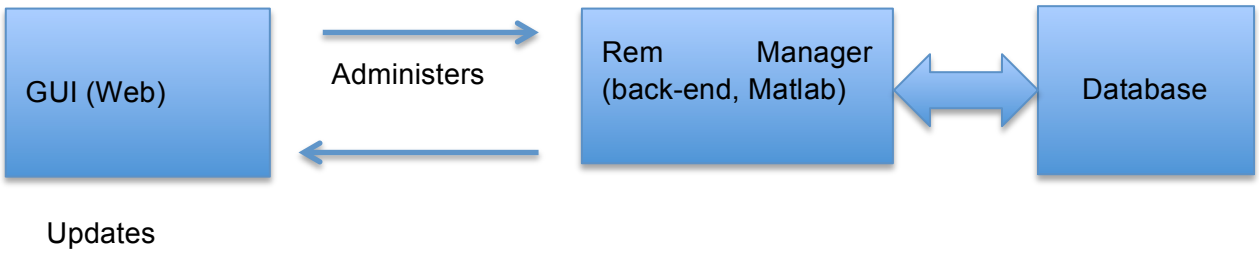


Figure 5: REM architecture.

Once decided the block structure and functions, what was left to do is to define the implementations for the functional blocks.

REM manager (Matlab):

Matlab is used to process and to create interpolated grids from the available channel measurement data samples concerting the primary user activity. Matlab provides an extensive set of mathematical functions, and file system utilities. More specifically, it has a native-implemented scattered sample interpolation library that performs excellently, and has a variety of algorithms available. Using Matlab for this block eased the burden of implementing an interpolator in some compiled language to perform the REM construction tasks. Also it made possible to develop benchmark for the various scattered interpolation algorithms available, in order to pick the optimal one for the REM construction.

REM manager (back-end):

The back-end contains the server logic for the REM manager. Its tasks include managing cognitive devices, keeping the database up to date and also updating and serving the webpage that acts as the GUI. This block is implemented using Node.js.

Node.js is a Javascript framework for developing the server part of web applications. It also features an extensive library of modules, which allow for implementing geospatial-processing applications, such as this project.

At the start of the project this part of the REM manager was to be implemented using Google Maps Engine, a Google service for managing geospatial data, but, unexpectedly, Google announced that Google Maps Engine would be discontinued and a dedicated back-end software module had to be created to comply with the project requirements.

Database:

The storage has to perform with two functions. Store the interpolated grid data that contains the power readings of the primary users at various locations, and maintaining an up to date database of the cognitive devices.

The first type, that references primary users activity is mostly static information, because of the static nature of TV-WS, and can be stored simply as a file with a particular format, in this case, a GeoJson. JSON is a particular data format that is widely used as a data interchange format in web applications. GeoJson is a standard json format used for geospatial applications.

The second type of information, the cognitive device list is not static, so it is better stored in a database. The database chose in this case is a sqlite3 database. It can be accessed using standard SQL commands, and it is auto-contained, making the deployment of the application easier.

GUI:

The function of the GUI block is to provide an easy way to consult the REM information. The GUI is developed as a webpage, which uses Google Maps API to provide geographical data, and Javascript functions, to show the primary and secondary user activity as overlays to the map.

The GUI also offers administration features. The GUI allows introducing simulated devices to the REM, at locations selected by the user, to delete devices from the REM and to change relevant parameters used by the REM, such as the grid size for the primary user activity data, or the information about the operational parameters of the cognitive devices that form the network.

Last of all, the GUI is synchronized in real time with the back-end, so any change to the REM, will be synchronized to the GUI.

2. Interpolation selection and implementation:

Once Matlab was selected for this task, several methods for performing scattered interpolation were available through the ScatteredInterpolant class. This class implements various methods of scattered interpolation natively.

The methods available are the following:

- Nearest Neighbour interpolation.
- Natural Neighbour interpolation.
- Linear Interpolation.

To test the relevant performance metrics of each interpolation method in a REM a synthetic benchmark was developed.

The benchmark:

In order to evaluate each interpolation algorithm a synthetic benchmark was proposed, based in the one conducted on [8]. In it, a synthetic power distribution simulating 3 radio sources located at different positions emitting in the same channel, and with propagation loss based on flat earth model is used.

In the experiment, a number of random scattered data samples are extracted from the synthetic distribution, and fed to each interpolator. Then, the interpolator has to reconstruct a grid from the sample data and the resulting grid is compared against the original synthetic distribution.

In order to interpret correctly the results of the benchmark, it must be remembered that the purpose of the REM is to detect the primary users, in order to not interfere with them and to detect any spectral holes available. In order to decide if a primary user is transmitting, or not, a detection threshold is used. If the power reading is above the threshold, the channel is flagged as used, or if below, the channel is flagged as empty.

Then, in order to correctly evaluate the benchmark the following three metrics must be evaluated:

As the interpolated grid differs from the original, there can be grid points where the interpolated power level is above the detection threshold, and the real power level is below it. That event is called a false alarm. In that case a site that could be used for transmission is wasted because the primary user is falsely detected.

Another more dangerous event is not detecting the primary user. Complimentary as before, some point of the grid could be below the threshold, being above the threshold in the real power distribution. This possibility is really dangerous because it leads to primary user is being interfered.

Finally other useful figure is the RMSE between the synthetic distribution and the interpolated one, because it gives a degree of closeness between the two figures.

For evaluating each interpolation method the three parameters described above as figures of merit were used, and the results are reported in section 4.4 of this document.

Implementation:

This functional block is tasked with reading the data samples from different files, creating a grid resulting from the interpolation of the initial scattered samples, and formatting the information from the grid to be used in the web application blocks.

Concretely, what the program does, is reading the sample information from a certain folder, and then, outputting a set of CSV files, accompanied with VRT files that are used to create the GeoJsons used in the web application.

Procedure:

Firstly, the basis for this project were a series of data from the TV spectrum occupancy taken in various points inside Barcelona metropolitan area, in two prior degree thesis.

The data was organized in data points, each of them referenced five different files, that contained the GPS location, the date in which the sample was taken, a vector containing spectrometer power readings, a vector with the frequency axis for the readings, and finally, a vector containing the power levels for each TV-channel.

For the construction of the REM the only relevant fields were the GPS position, and the TV channel levels.

The first step was to create a script that, from a folder that contained the various readings, extracted all relevant information and pulled it into a Matlab structure, that contained the point name, the position and the channel readings.

[data] = function loadData()

Now, the next step is using this data to interpolate a grid that contains the power readings for each channel, but first, the GPS data must be projected in order to translate the spherical coordinates to a set of Cartesian points.

Projection is a necessary step, because interpolation cannot be done in angular coordinates, since Euclid distance, needed for the correctness of the results of the interpolation algorithm, is not applicable in wgs84 coordinates (GPS) as they represent points in a sphere. For that purpose the UTM coordinate system was chosen, since it has low distortion for projections of zones inside the same UTM grid zone, and the metropolitan area of Barcelona is located entirely in UTM zone, 31T.

A Matlab function that transformed the data points GPS coordinates into UTM coordinates was developed, for that, the Geodetic Transformations Toolbox, developed by Peter Wasmeier, and available in the Matlab file exchange, was used.

Function [coordinates] = convert_GPS_values_utm(gps_coordinates)

After that, the next task was to interpolate the grid. In particular, a function that given the data positions in UTM coordinate system, and the channel readings, generated an interpolated grid for each of the TV channels was developed.

function [WGS_polygonal_matrix, WGS_Samples_matrix] = function process_all_data(data, grid_size)

This function takes as input parameter the separation between grid samples, in km. The grid size is chosen so the resulting square contains inside the convex hull of the data points.

Next the interpolated data must be formatted before being converted to GeoJson. In particular, each cell of the grid had to be expressed as a polygon for correct representation at the GUI. The polygon was generated in UTM coordinates and then translated al to elliptic, using the WGS84 ellipse for usage in the GUI. That polygon corresponds to the first column of the output CSV file and containing the geometry of the object to represent. The interpolated values of every channel are then concatenated to this string and form the rest of columns of the CSV file that is temporarily stored as a cell of strings, where each string corresponds to a CSV row, and a grid point.

The obtained interpolated data has also to be processed. First, the grid points that fall outside of the convex hull delimited by the data cannot be interpolated because they are too isolated, thus, the interpolator outputs NaN, or not defined, for the interpolated

values in this points. The CSV rows containing NaN values are then pruned from the cell array.

Next, a function to save the results to files was created and it is used to print the cell array containing the CSV rows values to a CSV file, and generate a companion VRT file. A VRT file is used as a description for a CSV file. It consists of XML encoded data that defines the contents of column of the CSV file. The VRT descriptor file is needed to convert the CSV file to GeoJson.

function save_all_data(sample_data, geometry_data)

Finally, after all the process, the GDAL open source utility is used to convert from the CSV files to GeoJson using its VRT descriptors. An external program was selected because a suitable implementation in Matlab was not found, and the actual conversion to GeoJson can take a long amount of development time to implement. GDAL is implemented in C, and actually faster than any Matlab script so using this program is not a performance concern. The program is called in the same Matlab script, using the system command, and then the newly generated files are moved to the public folder of the web-app to be served by the back-end.

4. REM Manager (back-end):

This software was written as a Node.js based web application and consists of two sub-sections. One is a normal web server that serves the webpage that acts as the GUI and other static content such as the GeoJsons that contain the primary user information. The other part consists in a REST API that is used to access REM functions such as adding devices, getting or deleting cognitive devices and administrating the REM working parameters. The back-end is also tasked with updating the GUI information at real time, using sockets.io.

Implementation:

The back-end is implemented using the express framework that is a simple and lightweight framework for creating web-apps for node.js. Express was chosen because it automates the most tedious processes in building a web-app or a REST API.

All the code is divided in modules. In Javascript, modules are parts of the program that reside in different source files. The advantage of this approach is it permits to compartmentalize the code in more maintainable units, than using only one source file.

Node.js usually performs asynchronic operations such as reading or writing intro files or databases. In order to keep the execution flow, the asynchronic methods admit a callback function as a parameter, which is called when the method finishes executing. This characteristic marks the style of the code, that normally features callbacks in each operation.

The functionality of the backend- is split into the following modules:

app.js -> initializes the http server, and binds the routes, also sets up the generic error handlers. Loaded at app start.

REM.js -> represents the model for the REM. Contains all the methods used to access and modify the REM database. Its methods are called from the API, to modify the contents of the REM.

REM.js manages an sqlite3 database, with a single table called devices, where every row corresponds to a device. The devices table has the following columns:

1. **device_id:** a numeric identifier for the device
2. **lat:** latitude in decimal degrees.
3. **lng**: longitude in decimal degrees
4. **tx_power:** the maximum EIRP for the device, in dBm.
5. **coverage_radius**: an approximated coverage radius.
6. **channels**: the channel indexes assigned to the device.

The methods exposed by the REM module are the following:

- **addDevice(lat,lng,bandwidth,callback(err,data))**
  - Tries to insert a new device in the network using the insertDevice(lat,lng,bandwidth) method from the spectrum-broker module. If the insertion is successful, stores the new device into the devices table of the database, and passes the device to the provided callback. Callback signature is callback(err,data).

- **deleteDevice(device_id,callback(err,data))**
  - Tries to delete the device that has the dame device_id property than the device_id argument. If successful, executes the callback.

- **deleteDevices(callback(err,data))**
  - Wipes completely the devices table, then executes the callback.

- **getDevices(callback(err,data))**
  - Obtain all the devices stored in the devices table. Retuns a vector with all the stored device objects and passes it to the callback.

- **updateDevices(callback(err,data))**
  - Called when a REM parammeter is changed. retrieves the devices table, and then then recalculates all devices properties, and updates them in the database. When finished, passes the updated devices to the callback.

spectrum-broker.js -> Is the module that assigns **channels** and **tx_power** to the new devices submitted to the REM. It uses the REM data to allocate the new devices and minimise interference for the primary users. This functionality was not required at the start of the project and is implemented in a basic way, however it is a fine way to showcase the REM capabilities.

The methods exposed by spectrum-broker are listed below:

- **insertDevice(lat,lng,bandwidth,callback(err,data))**
    - This method is the responsible of adding new devices to the network. The algorithm is actually quite simple. The pseudocode is found below:

        1. The first thing it does is to locate what cell in the grid contains the designated point for the new device. If the device sits outside the grid, returns an error.
        2. Next, checks if the bandwidth selected is allocable at the selected position. Normally, a series of available channel blocks will result. If no suitable channel block is found, an error is returned.
        3. The next step is selecting the channel block, it calls the **selectChannels** method, of the same module, and that will output the assigned channel block.
        4. Once the channel block is chosen, the max transmission power for the device is selected. It is obtained from the **getMaxTxPow** method.
        5. The final step is to calculate the approximated coverage radius for the device. for this task, the method **findCoverage** is used.
        6. Once all the parameters are set, a new Device object, with the obtained transmission power, coverage radius and selected channels is passed to the callback function and the execution ends.

- **selectChannels(num_channels, available_blocks)**
    - This function just selects the lowest block of num_channels length possible.
- **getMaxTxPow(insertion_point,block)**
    - The objective of this function is to find the maximum Tx_power for the device that will not degrade the CIR for the primary users below a certain threshold. It assigns the entire available interference margin to the device if needed.

- The maximum CIR that primary users can affront is a parameter of the REM:

$$CIR = \frac{S}{N + I}$$

Let sEIRP be the EIRP of the secondary user at the direction of the primary user antenna and let L be the path loss, obtained using a flat earth model, with exponent alpha is:

$$I = sEIRP * L$$

$$L = \frac{h_b^2 * h_m^2}{d^\alpha}$$

Then, the sEIRP for a given CIR ratio constraint can be found as:

$$sEIRP = \frac{\frac{S}{CIR} - N}{L}$$

And if hypothetic secondary users with isotropic antennas are used, the max transmission power can be equated to sEIRP.

This formula is used in the pseudo code below to get the max transmission power achievable without breaking the interference constraints for a single device.

```
TxPow = MaxTxPow
    for every channel in the block
        for every cell in the grid
            if cell is used by primary
                obtain the distance between the insertion point and the cell
                find tentativeTxPow for the selected distance
                if tentativeTxPow < TxPow
                    then TxPow = tentativeTxPow
                end
            end
        end
    end
return TxPow
```

- **getCoverage(pow)**
  - Just returns the approximate coverage radius, considering the interference level at the grid cell where the device resides and a flat earth path loss model.

Device.js -> this module does not implement any method, just defines the Device object, and some static properties used by the spectrum broker module:

- **Device Height**: hm in flat earth model.
- **Base Station Height:** hb in flat earth model.
- **MaxTx:** Defines the max transmission power for the device, in dBm.
- **SCIR:** devfines the minimum secondary user CIR.

api.js -> Defines the route used by API commands and the functions called by each HTTP verb. The functionality and the structure of API calls is described at the appendices.

params.js -> Defines the route and the commands used to modify the REM parameters. The functionality is defined in the appendices.

All this operations also update the web-connected client GUI in real time using sockets.io, so changes in the cognitive devices list, or in the REM parameters are immediately updated.

5. Front-end:

The front-end webpage interface is based in the Google Maps Javascript API. This API allows the user to paint overlay objects to the map.

The web consists on a HTML, with a full-page map view, and several boxes containing information, input and controls.

The implemented Javascript code (around 600 lines in length) performs functions such as updating the view contents, displaying the map data overlays and communicating with the back-end.

The operation of the scripts follows the loading of the page, sequentially:

1. The browser loads the HTML, and the Javascript files.
2. When the page and its contents are loaded the load finished event is triggered, which calls the function initialize().
3. The function initialize() loads the Google Maps window, and the geoJsons from the backend, creating a map view of the Primary User information.
4. Queries the backend API for the device list using a GET request to the route “/api” and “/params”, this loads the Secondary User information and the configured REM operational parameters.
5. Opens a socket.io connection with the backend.

Once this process has ended, the webpage will display the availability grid for a certain channel, with markers showing the devices added to the REM, and their approximated coverage radius, if some of the channels used by them are selected.

If some change is made in the REM, either by the GUI user, someone using the API, or another GUI user connected to the REM, the socket connection will make the view update in real time. That applies also if the REM parameters are changed.

The frontend offers the user the following functionality:

- Visualization of the data contained in the REM, the primary user power levels grid by selected channel, and the cognitive devices and channel availability for a selected location.
- An interface for visualizing and modifying the REM parameters.
- A convenient way to add and delete simulated devices to the network.

The communication with the server is achieved using AJAX and Sockets.io.

AJAX is a javascript library used for sending http requests, and handling the responses asynchronously, using callbacks. Heres an example of an AJAX operation. In this case an object containing changed parameters is sended to the backend using method POST. There is no callback defined to be called when the response arrives.

```
$(document).ready(function () {
  $.ajax({
    type: 'POST',
    url: 'http://localhost:3000/params',
    dataType: 'json',
    contentType: "application/json",
    data: JSON.stringify(payload)
    });
  });
```

Here is an example of another AJAX operation. In this case a successful response executes a callback that can operate with the contents of the response.

```
$.ajax({
    type: 'GET',
    url: 'http://localhost:3000/api',
    success: function (data) { // …
```

Sockets.io is a real time connection oriented communication library over http. The information exchange is event based. To use it, one has to open a socket connection using:

```
var socket = io();
```

Then, the next step is setting callbacks for the different events received.

For example, the backend will broadcast to the browser clients a “device added” event when a device is added through the API. To receive the event, the client side script implements the following code:

```
socket.on('device added', function(msg){
        devices.push(new Device(msg));
    });
```

This snippet of code sets a new callback on the socket connection opened before. When a “device added” event arrives, a new device will be created from the contents of the event, and added to the client devices list.

The connection can be also used to send events from the browser client:

```
socket.emit(‘event name’, {msg: hello});
```

This would generate a ‘event name’ event at the backend, with content {msg: hello}.

4. Results:

Despite the setback suffered when Google announced that Google Maps Engine would be closed, all objectives and requirements of the project were completed, even the extension objective of creating a Demo app for the REM.

4.1. GUI:

The resulting GUI allows the user to consult REM information easily. In [figure 6] a sample screenshot of the GUI is represented.

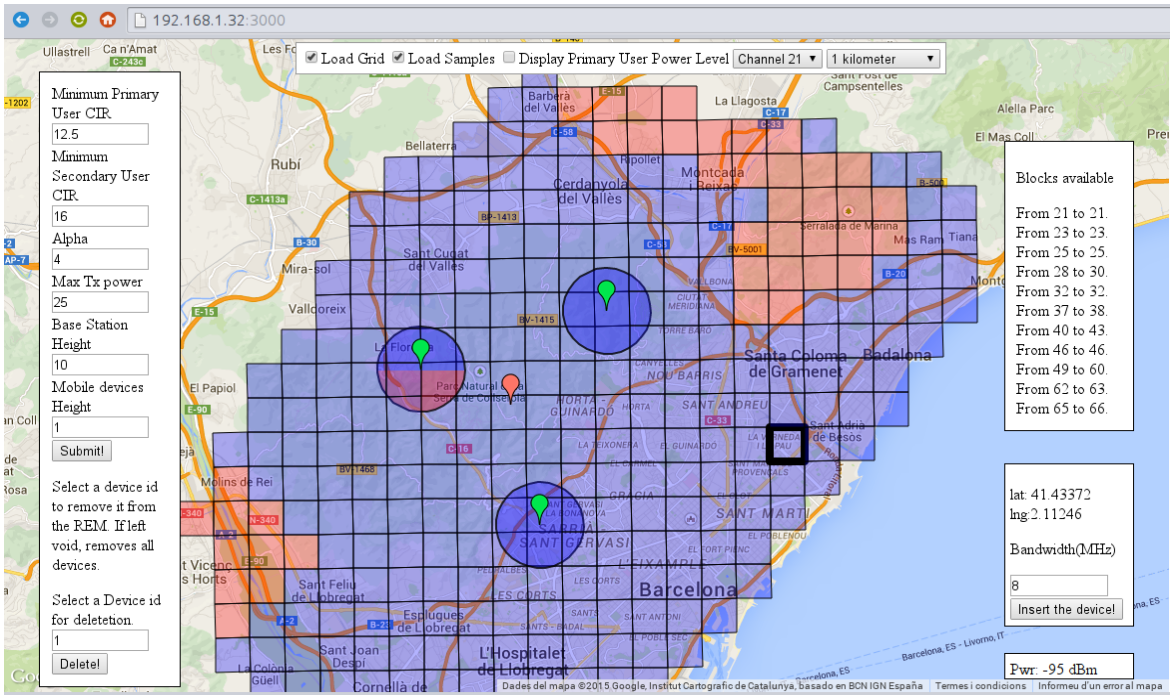


Figure 6: sample GUI screenshot.

The primary user data is represented as a grid containing the primary user power readings for each channel. The grid is painted red if the power reading for Primary Users is above the detection threshold, in this case, -85 dBm, and blue otherwise, indicating that the this channel in this cell can be accessed. The visualized channel can be changed using the toolbar located at the upper edge of the GUI. This tool allows the GUI user to check the geographical availability of each TV-WS.

The Primary user readings can also be observed as a heat map, if the user click the checkbox Display Primary User Power Levels. The colour styling of the grid will change into a gradient representing the detected PU power level. An example of this view is shown at [figure 7].

The markers in [figure 6] and [figure 7] represent cognitive devices introduced in the network. The markers will be shown green, with an approximate coverage circle if the devices are using the selected channel. In case they are not using it, the markers will be shown grey [figure 8]. If the user hovers the mouse over a device marker, a tooltip containing information about the device in question will appear, as shown in [figure 8].

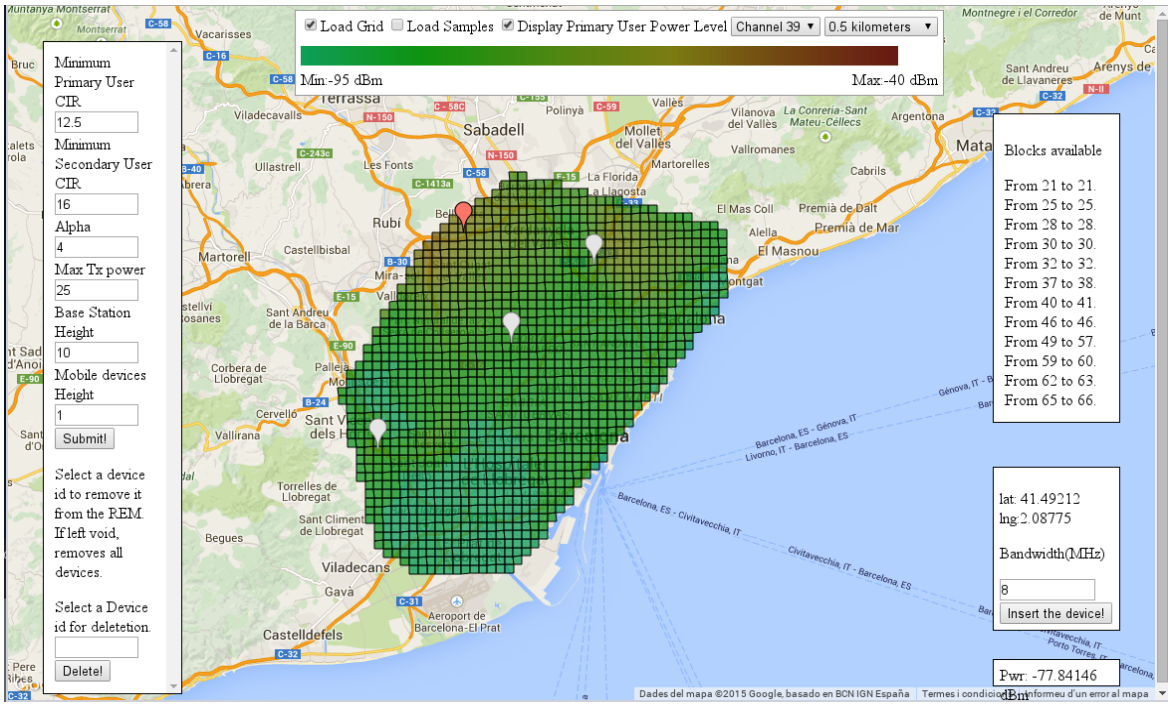


Figure 7: sample GUI heatmap visualization.

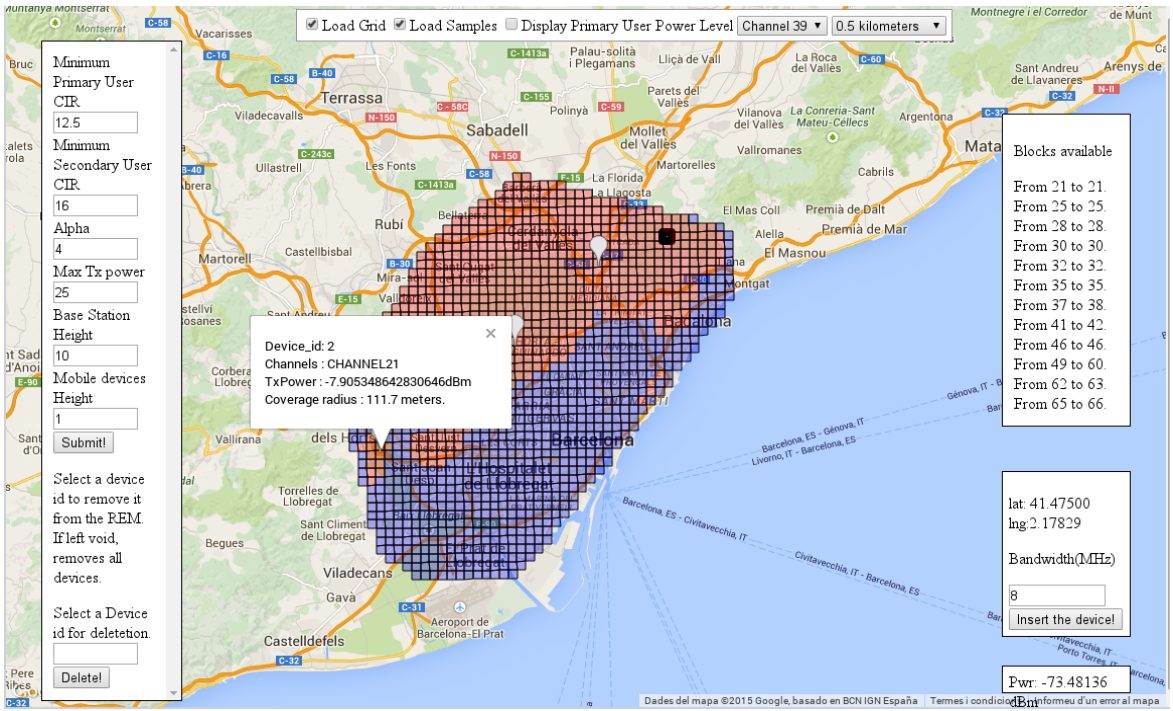


Figure 8: devices operating outside the selected channel

The GUI user can also click on a certain point to obtain information on the available channel blocks. If the user clicks on a certain point inside the grid, a red marker will appear on the selected location, and on the left edge of the screen the geographical coordinates and the available channel blocks in that location will be represented. This behaviour can be observed in [figure 6] and [figure 7].

On addition, the GUI can also be used to insert simulated cognitive devices in the network, deleting them and changing the parameters the REM uses to insert devices, or selecting different grid sizes for the Primary User information.

If the parameters of the REM are changed, or the grid size changes, all devices are updated and reintroduced in the REM. This procedure must be done since the information used to insert the devices into the REM has changed.

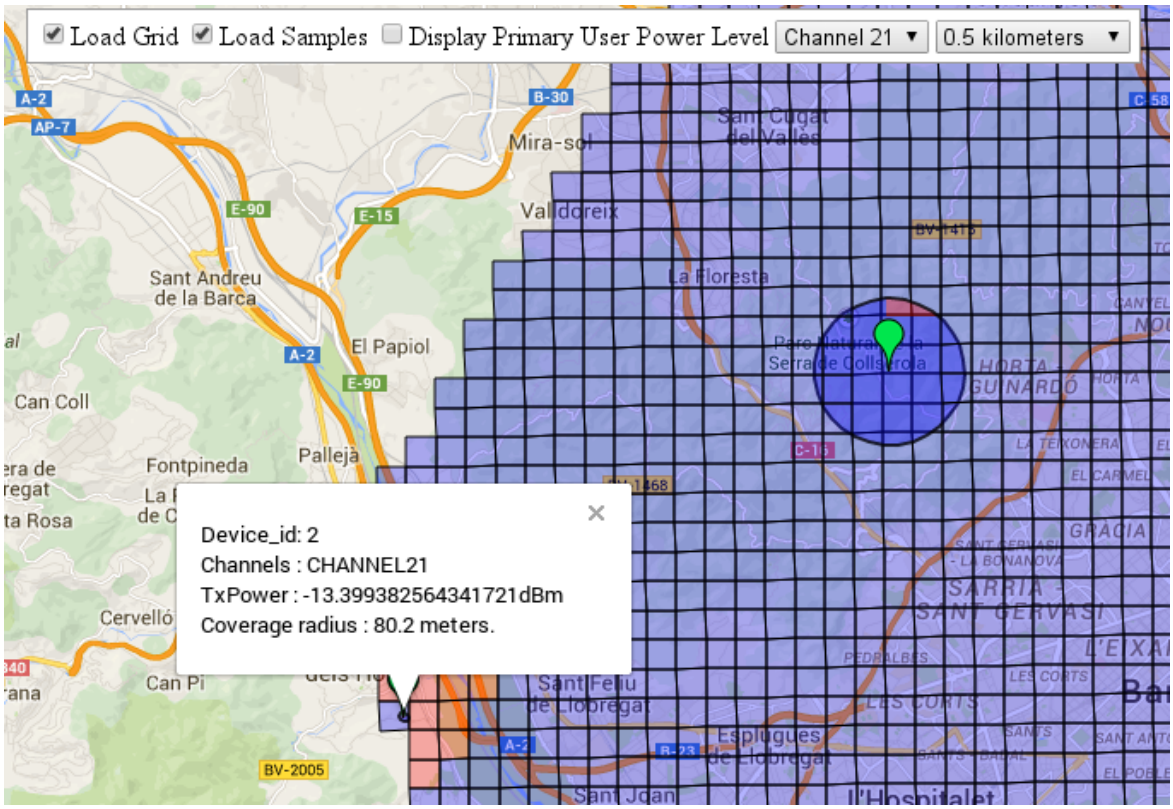


Figure 9: Device 2 in the original 500m grid.

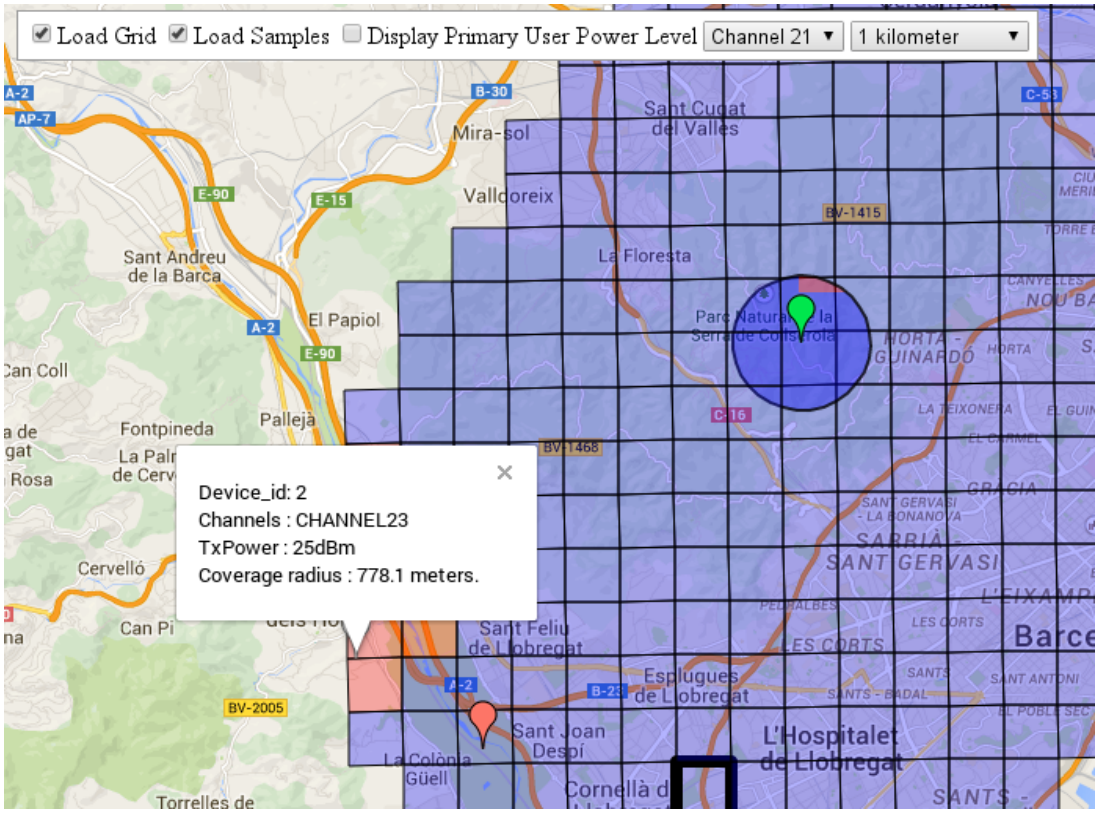


Figure 10: Device 2 in the updated 1km grid.

It is important to note that all changes in the GUI are synchronized with the REM manager in real time, and also between different GUIs connected to the same REM manager. The effect of this feature is that if there were more than one user connected via web, the changes to the REM done in one of the GUIs would be updated to the other GUI immediately.

On the particular case of a grid resolution change, reintroducing the devices in the REM is especially important, since the geographical availability of the channel will actually change. In zones where the signal level is close to the threshold a finer granularity on the interpolation allows to approximate with more precision the activity of the primary users.

It is important to note that the minimum 250 meters grid resolution is due to performance constraints on the GUI side. The test machine was a 2009 Thinkpad t400 with a core2duo @ 2,13 GHz, running Ubuntu Linux and on Chromium browser.

[Figure 9] and [Figure 10] show the result of a grid size change. In them we can observe that the parameters for the selected device has changed. The device with device_id 2, was using channel 21 with the 500m grid, and when the grid size changed to 1km the zone where the device sits is marker ad occupied by the Primary user. The device is then introduced in the channel 23, where that cell is marked as free in the new grid.

4.2. REM Manager:

The two sub-components of the REM Manager, the back-end and the Matlab script perform correctly, within the specifications. However it is interesting to disclose the performance constrains of the system.

For the Matlab script the total run time for generating the three used grids was 3 minutes, on the same PC used for the GUI testing. It might seem like a bottleneck for the system at first glance but, as the Primary User activity is almost static, the task of creating new GeoJson files has only to be performed seldom.

The performance of the back-end is also not a concern for the grid sizes and algorithms used. The median time for a device insertion is around 20 ms, and its almost constant for every grid size. This might seem counter intuitive, as for a greater number of cells, the difficulty of introducing a new device increments. A possible explanation for this phenomenon is that what really impacts the run time of this operation is the introduction of the device in the database, so the bottleneck for this system, is I/O performance.

The run time of the most computationally intensive operation, updating the whole cognitive network after a parameter change, has a run time that seems to be dependent of the number of devices. This also correlates with a system limited by the I/O operations as more devices on the network means more update operations for the database, which are performed sequentially. For reference, the run time of a update for 20 cognitive devices is close to 500 ms.

4.3. Demo App:

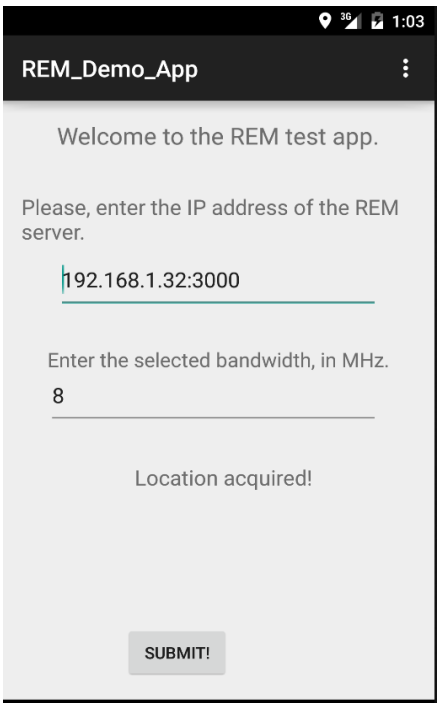
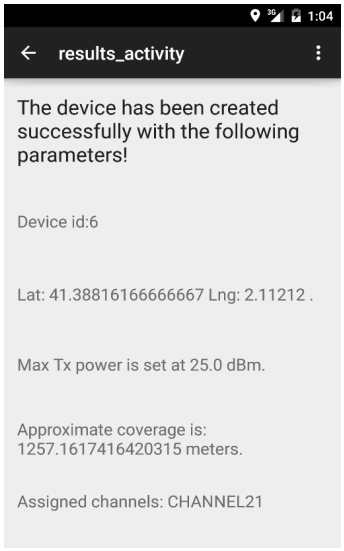


Figure 11: app interface

The demo application is an android app which uses the device geo-location features and a bandwidth parameter chosen by the user to insert a new simulated cognitive device through the REM manager API. [Figure 11] shows the interface of the test app for Android 5.0.1.

Once the REM manager responds to the query, the application shows the received information, or error, for the newly introduced device, shown on [Figure 12].



This demo shows that the requirements to use cognitive access through a REM are very limited; a device only needs a way to access an approximate location, and an http client to use the prototype REM described in this thesis.

Figure 12: App results

4.4. Interpolation algorithm validation:

The last objective of the thesis was the validation of an algorithm for REM construction, for that task, the synthetic benchmark described at section 3.2 was used. The benchmark was iterated several times for greater statistical significance, and also tested with different data sample sizes.

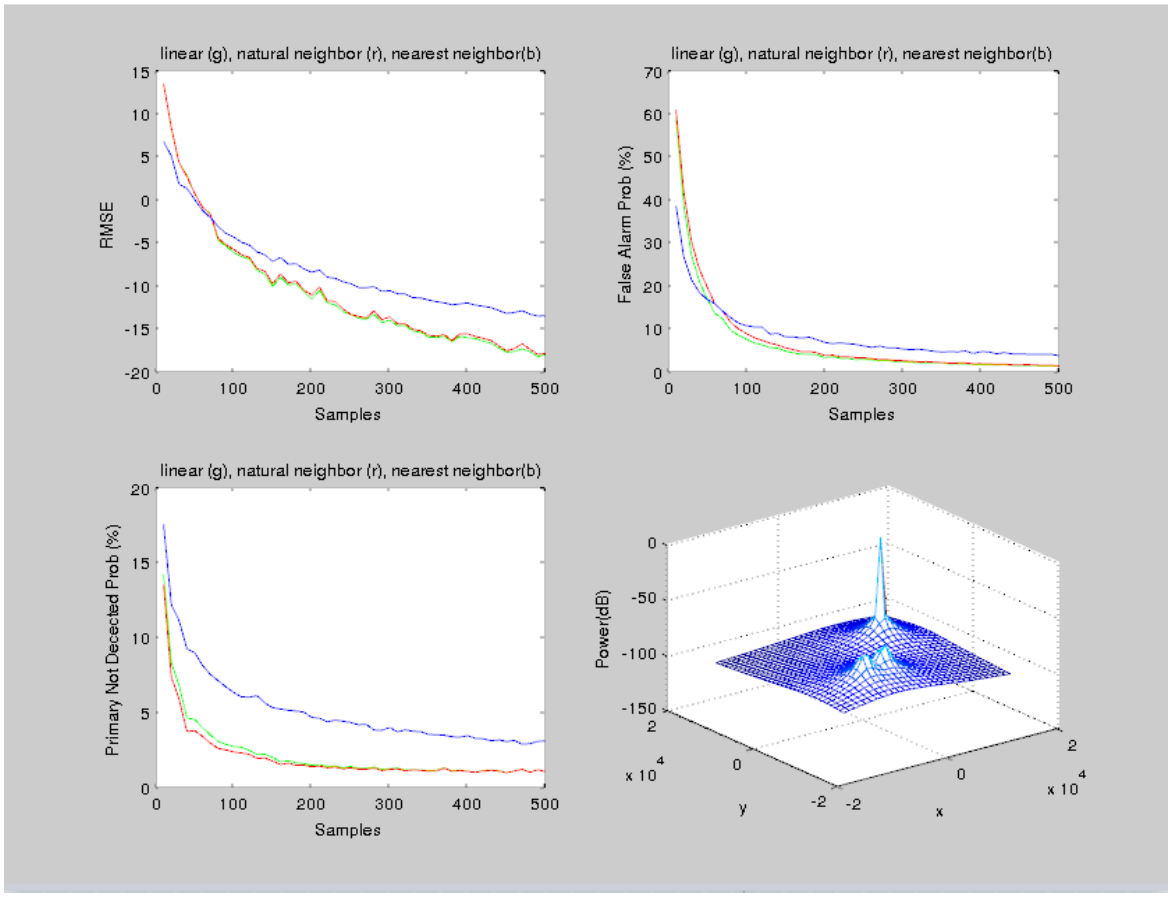


Figure 13: results of the synthetic benchmark.

The results in [Figure 13] show the relevant figures of merit of the diverse interpolation algorithms versus the number of data samples used in the interpolation. The lower left corner figure, shows the original synthetic power distribution, in dBw.

The RSME figure shows that, for a moderate amount of samples, higher than 100, linear interpolation and natural neighbour interpolation perform almost identically, with a marginal advantage to linear interpolation.

The figure for False Alarm Probability show the same tendency, benefiting nearest neighbour interpolation for a small amount of samples (detail at [figure 14]), and linear and natural neighbour interpolation performing almost equally for large amounts of samples.

Finally, in the probability of Primary not detected graph, it can be seen that Natural neighbour interpolation performs better than the other algorithms, especially for a limited amount of samples, (detail at [figure 15]) . Detecting the primary user correctly is the key parameter for a cognitive radio, as cognitive radios operate on the premise of minimizing interference with Primary Users.

From the results of this test, and taking into account the prevalence of Primary User detection in the operation of cognitive radio, the algorithm that was chosen to construct the REM data was Natural Neighbour Interpolation, as it performed better or equally detecting the primary users for all sample sizes.

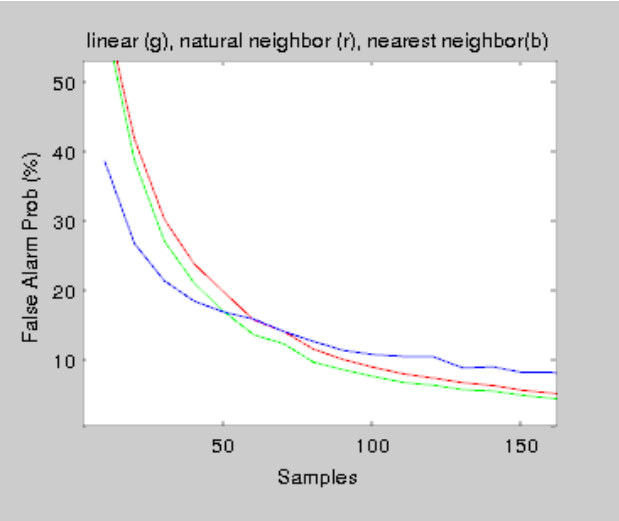


Figure 14: Alarm Probability versus sample size .

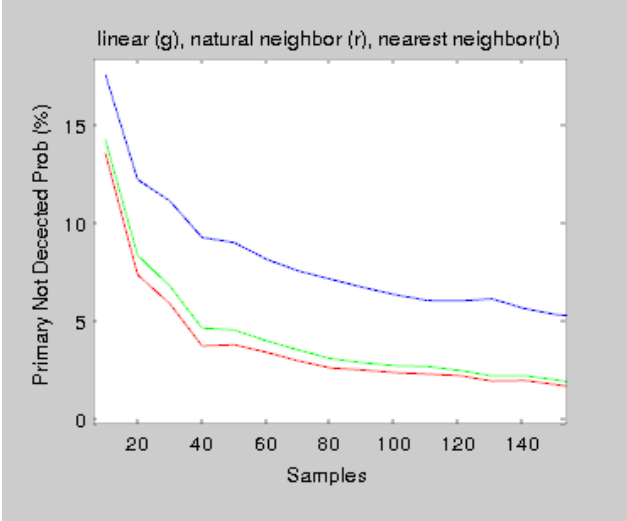


Figure 15: Primary Not Detected versus sample size.

5. Budget:

5.1. Labor costs:

The personnel cost table below is based on the following figures:

- A junior engineer perceiving 13.04 Euros per hour.
- The employee costs a 30 % more to the employer than the perceived remuneration due to social security and other taxes, elevating the cost per hour to 17,4 euros.

Task	Dedicated Hours	Cost (Euros)
WP 1: Planning and research	37	643.8
WP 2: Rem architecture definition	15	261
WP 3: Interpolation methodology	47	817.8
WP 4: Creating a GUI	55	957
WP 5: Integration and validation	122	2122.8
WP 6:Creation of a mobile demo	10	174
WP 7:Final Report	75	1305
Total	**361**	**6281.4**

Table 8: Labour costs.

5.2. Software costs:

The only non-free software used in the development of this project, was Matlab for unix, systems. The cost is described below.

Matlab license for education
Yearly cost: 500 Euros
Used for 143 days
Cost: 196 Euros

Table 9: Software costs.

5.3. Amortizations:

The development of this project only used a single laptop computer for development tasks.

Laptop for the developer
Purchase cost: 900 euros
Amortization period, 4 years, residual value, 25%.
Used for 143 days.
Depreciation: 88.15 euros.

Table 10: Depreciation.

6. Conclusions and future development:

6.1. Summary:

The technical result of this project is clear. A REM for cognitive utilization of TV-WS can be implemented through generic technologies in a cost effective manner, using, mostly, open source tools.

On a personal level, completing this thesis has helped me to develop my skills in two key areas. The first of them is bibliographical research, which has been key to gather sufficient information, in a timely manner in order to develop this thesis. The other area in which my knowledge has improved is web development. At the start of this thesis my knowledge of web technologies was superficial, at most, but during the development of the project I learned about front-end development, back-end development and database management, all done in a programming language I was not very proficient at the start of the project, Javascript.

6.2. Future Development:

The product of this thesis can be expanded in the following ways:

1. Extending the functionality of the spectrum broker module. Now the spectrum broker uses a pair of basic functions to introduce cognitive devices in the network, and does not take into account the interference between the devices. Also, it does not optimise the assignment of the spectrum for the cognitive users.
   In order to maximise the overall capacity of the network, this factors should be taken into account when allocating the devices on a certain channel, or assigning their maximum EIRP.

2. Implementing and validating an authentication scheme for accessing the REM and implementing privacy for the communications between the users and the REM manager. At its current state, all API operations are performed on plaintext, and without authentication, implementing this features would increase significantly the overall safety of the users, because the REM manager works with private data such as the cognitive users geographical positions, also it would protect the REM manager from being tampered with by unauthorized users.

Bibliography:

[1] Mitola, J.; Maguire, G.Q., Jr., "Cognitive radio: making software radios more personal," Personal Communications, IEEE , vol.6, no.4, pp.13,18, Aug 1999. DOI: 10.1109/98.788210

[2] Haykin, Simon, "Cognitive radio: brain-empowered wireless communications," Selected Areas in Communications, IEEE Journal on , vol.23, no.2, pp.201,220, Feb. 2005 .DOI: 10.1109/JSAC.2004.839380

[3] Mitola III, J. Cognitive Radio: An integrated agent architecture for software radio architecture. Diss. Ph. D Dissertation, Royal Institute of Technology (KTH), 2000.

[4] Jondral, F.K., "Cognitive Radio: A Communications Engineering View," Wireless Communications, IEEE , vol.14, no.4, pp.28,33, August 2007 DOI: 10.1109/MWC.2007.4300980

[5] Lopez-Benitez, M.; Umbert, A.; Casadevall, F., "Evaluation of Spectrum Occupancy in Spain for Cognitive Radio Applications," Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th , vol., no., pp.1,5, 26-29 April 2009. DOI: 10.1109/VETECS.2009.5073544

[6] van de Beek, J.; Riihijarvi, J.; Achtzehn, A.; Mahonen, P., "TV White Space in Europe," Mobile Computing, IEEE Transactions on , vol.11, no.2, pp.178,188, Feb. 2012 doi: 10.1109/TMC.2011.203

[7] Marques, P.; Rodriguez, J.; Delaere, S.; Delahaye, P.; Lecroart, B.; Gundlach, M.; Triantafyllopoulou, D.; Moessner, K.; Noguet, D., "Spectrum sharing in the EU and the path towards standardization," Future Network and Mobile Summit (FutureNetworkSummit), 2013 , vol., no., pp.1,9, 3-5 July 2013

[8] Zander, J.; Rasmussen, L.K.; Ki Won Sung; Mahonen, P.; Petrova, M.; Jantti, R.; Kronander, J., "On the scalability of cognitive radio: assessing the commercial viability of secondary spectrum access," Wireless Communications, IEEE , vol.20, no.2, pp.28,36, April 2013. DOI: 10.1109/MWC.2013.6507391

[9] Yilmaz, H.B.; Tugcu, T.; Alagöz, F.; Bayhan, S., "Radio environment map as enabler for practical cognitive radio networks," Communications Magazine, IEEE , vol.51, no.12, pp.162,169, December 2013. DOI: 10.1109/MCOM.2013.6685772

[10] Farnham, T., "REM based approach for hidden node detection and avoidance in cognitive radio networks," Global Communications Conference (GLOBECOM), 2012 IEEE , vol., no., pp.1391,1397, 3-7 Dec. 2012. DOI: 10.1109/GLOCOM.2012.6503308

[11] Jian-Wu Zhang; Qi Zhao; Jing-yuan Zou, "The IEEE802.22 WRAN System Based on Radio Environment Map (REM)," Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on , vol.1, no., pp.98,101, 7-8 March 2009. DOI: 10.1109/ETCS.2009.31

[12] Youping Zhao; Reed, J.H.; Shiwen Mao; Bae, Kyung K., "Overhead Analysis for Radio Environment Mapenabled Cognitive Radio Networks," Networking Technologies for Software Defined Radio Networks, 2006. SDR '06.1st IEEE

Workshop on , vol., no., pp.18,25, 25-25 Sept. 2006. DOI: 10.1109/SDR.2006.4286322

[13] Stevenson, C.; Chouinard, G.; Zhongding Lei; Wendong Hu; Shellhammer, S.J.; Caldwell, W., "IEEE 802.22: The first cognitive radio wireless regional area network standard," Communications Magazine, IEEE , vol.47, no.1, pp.130,138, January 2009. DOI: 10.1109/MCOM.2009.4752688

Appendices:

GUI User's manual:

The purpose of this section is to describe the functionality of the GUI for the REM. Latter, a step-by-step guide explaining how to perform the various use cases for the GUI will be provided.

The content of the GUI is the following:

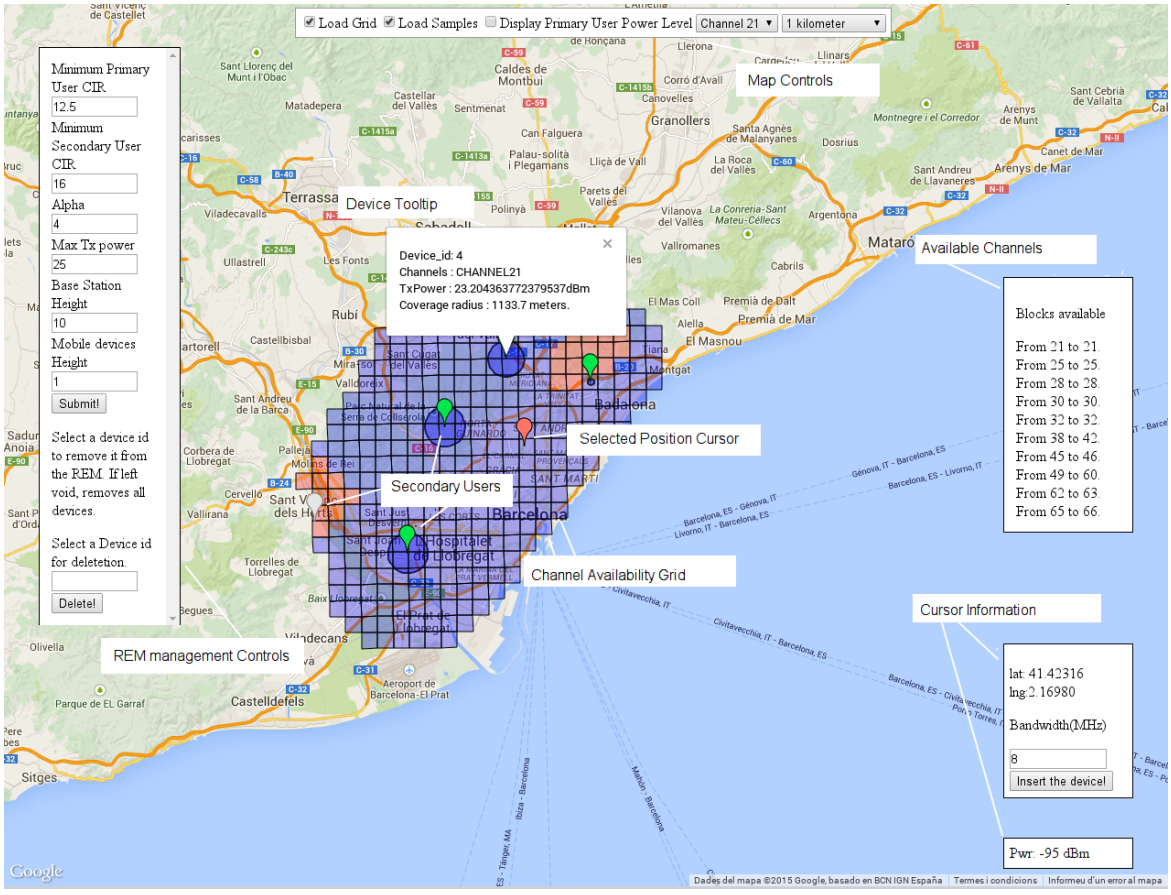


Figure 16: Elements of the GUI

The different sections are explained below:

- Map Controls: Contains the elements that control which information is displayed on the map.
- REM management controls: Allow the user to change the parameters that the REM uses to insert cognitive devices, and to delete devices from the REM.
- Selected Position Cursor: Allows the user to select one location from the map. This position controls the information shown at the Available Channels and Cursor Information elements.

- Available Channels: List the UHF channel blocks available for cognitive use at the cursor location.
- Cursor Information: Displays the primary user power level at the cursor's position and the position of the cursor. Also lets the user specify a bandwidth for creating a new simulated cognitive device at the cursor's position.
- Channel availability grid: It is a grid overlay that is used to display the information about the primary users behaviour in the area. The cell coloring can show two types of information:
    - If the Display Primary User Power Level checkbox, in the map controls is not checked. The color code will show if primary user activity is detected (red) or if such activity is not detected and the channel is available for cognitive use.
    - If the after mentioned checkbox is checked, the color code will represent the total power detected at each channel.
- Secondary users: Green or grey markers are used to represent the cognitive devices stored in the REM. A device will have only one of the two representations if the following conditions are applied:
    - If the device in question is using the UHF channel represented in the GUI, then a green marker, with a circle that marks its approximate coverage range, will represent the device.
    - If the device is not using the selected channel, a grey marker will represent the device.
- Device tooltip: It is a short description of the device main parameters, that appears when the user moves the mouse pointer over the device's marker.

Use cases:

- **Changing the displayed UHF channel on the map:**

  To change the displayed channel, the user has to select the desired channel in the dropdown selection menu, shown in fig[15].

Figure 17: channel selection.

- **Change the REM grid size:**

  Select a new grid size from available sizes at the dropdown menu highlighted in fig[16].

Figure 18: grid selection.

- **Change grid styling:**

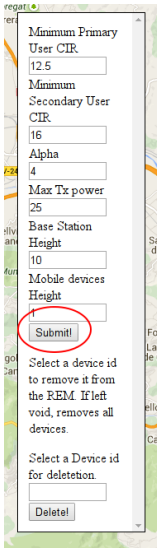
  Check the highlighted checkbox in [Figure 20] to switch between the grid colours representing cognitive availability, or primary user power levels.

Figure 20: grid style selection.

- **Display original samples, toggle grid visibility:**

  Uncheck the load grid checkbox to hide the grid overlay, or toggle load samples to overlay markers that show where the original channel measures used to construct the REM were taken.

Figure 21: display selection.



- **Change the REM parameters:**

To change the REM operational parameters, alter the desired fields, and click the submit button.

- **Delete devices from the network:**

To delete a device from the network, write the device_id property of the device to be deleted at the input above the delete button, (can be obtained from device tooltip) and click the button.

To delete all cognitive devices click the “delete!” button, leaving the input field empty.

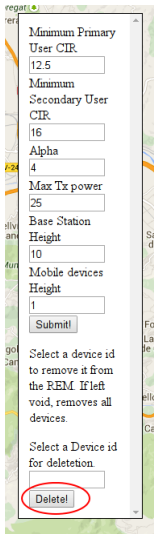
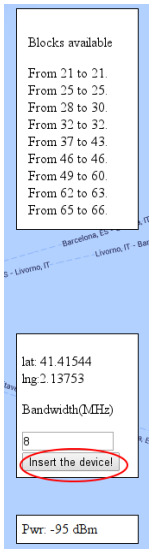


Figure 22: REM parameters

Figure 23: delete devices.



- **Add a simulated device to the REM:**

In order to add a new device, click on a point inside the grid to select a position. A red marker will appear at the selected position. The available channel blocks for cognitive will appear. Input a bandwidth, in MHz at the box above the “Insert the device!”, then click on the button.

Figure 24: Insert Device

API documentation :

The interaction with the REM manager is handled through a REST API, and with JSON as interchange format. The paths, and available interactions are listed below:

"/api" : used to give commands to the REM, different http verbs trigger different actions. All requests must have set the headers content: application/json, because all communication is done via JSON files.

POST: tries to add a device to the network, the request must carry a JSON with the structure:

{

"lat": xx,

"lng":yy,

"bandwidth": zz

}

Being lat, lng the coordinates of the device in decimal degrees, and bandwidth expressed in MHz.

If the request is valid, the response will be either the inserted device, if the device can be inserted correctly in the network, or a JSON containing an error field with a string identifying the error cause. Error causes can be that the device is outside the REM grid, or that the REM cannot allocate the selected bandwidth at the given location.

The device object returned will have the same fields as in this example:

{
"device_id": 27,
"lng": 2.143904,
"lat": 41.415651,
"tx_pow": 19.269792929719877,
"coverage_radius": 1045.3093215671413,
"channels": "CHANNEL21"
}

The tx_pow field indicates the maximum transmission power for the device at the selected location, in dBm. Coverage radius is an estimation of the coverage of the device and the channels field is a string with the name of the channels assigned for transmission.

In case of error, the response will be:

{

"error": " The device at [2.143904,41.415651] asks for too much bandwidth."

}

Containing a string related to the cause of the error.

DELETE: If a JSON with the field id is provided with the request, the device with the same device_id will be deleted from the network. If no json is provided, the backend just will delete all the devices on the network.

For deleting only a device, the querry JSON:

{

“device_id”: 3

}

the response for a successful querry is:

{

“message”: “ok”

}

“/params” is used to change the parameters that the spectrum broker uses to allocate the devices:

MinimumPCIR: is the minimum CIR primary user needs in order to operate.

Alpha: The path loss exponent used in the REM.

HeightB and heightM: the heights used in the flat earth propagation model.

MinimumSCIR: is the minimum CIR secondary user needs to operate.

GridSize: is the separation between samples in the REM grid.

The route admits the following verbs:

“GET”: returns a JSON with the values of all the parameters used, example response:

{

"minimumCIR": 12.5,

"alpha": 4

"maxPow": 25,

"minSNR": 16,

"heightB": 10,

"heightM": 1,

“gridSize”: 1

}

“POST” updates the parameters to the selected values, to update a parameter a JSON with the parameter name and the new value must be send:

example:

{

“alpha”: 2

}

The response JSON is a vector containing the updated version of the REM, recalculated with the new parameters, or an error if invalid information is passed.

Interpolation methodology:

Choosing a suitable interpolation methodology is an important factor in the construction of a REM. An adequate interpolation methodology will provide the REM with accurate estimations of the state of the channel in most geographical areas, enabling the cognitive network to make the best possible use of the channel, without interfering its owners.

The mathematical problem that the interpolator has to face is reconstructing a complete grid for the Radio Environment Map from a set of scattered (non-uniform distributed) values. In this case, the aim to reconstruct, or interpolate , the values inside the region , or convex hull, of the dataset. The solution must form a surface where two dimensions correspond to the geographical positions an the other to the field intensity, for each channel. To solve this problem, the Matlab class ScatteredInterpolator three different algorithms, that will be briefly explained below:

1. Nearest Neighbour: This is the simplest method, where an interpolated point is given the same value as the nearest data sample, using a certain definition of distance. It's the fastest from the three methods available, but also happens to be the least accurate.
2. Natural Neighbour: This method is based on the Voronoi decomposition of a set of points. This decomposition generates a partition of the plane where the point set is located, assigning each point a Voronoi cell. The Voronoi cell of a given point is the area of the plane that is closer to it than any other point of the set, given a certain definition of distance. When a new point is introduced in the set, or interpolated, the Voronoi diagram changes, as a cell must be assigned to this new point. This new cell will have an intersection with the cells that formed the diagram before its introduction, its neighbours. Finally, the value of the new introduced point is calculated as a weighted average of the values of its neighbours. The weights used are proportion of each neighbouring cell area, or “stolen” area, that has gone into the new cell containing the interpolated point.
3. Linear interpolation: This method uses a decomposition of the sample space in triangles. All the samples are joined in groups of three, creating a surface formed by flat triangular shapes, joined by their edges. Each of the individual triangles, defines a surface, and the parameters of that surface can be obtained , given the three points that form it. To interpolate a point, first, the point is located into one of the triangles that form the sample space, then, the value interpolated value for the point its the one that adjusts to the surface equation defined by the points in the triangle where is located.

Glossary:

CR: Cognitive Radio.

REM: Radio Environment Map.

PU: Primary User.

SU: Secondary User.

TV-WS: TV White Spaces.

DSP: Digital Signal Processor.

FPGA: Field Programmable Gate Array.

ASIC: Application-Specific Integrated Circuit.

REST: REpresentational Stateless Transfer.

API: Application Programming Interface.

GUI: Graphical User Interface.

HTTP: HyperText Transfer Protocol.

HTML: HyperText Markup Language.

JSON: JavaScript Object Notation.

SQL: Sequential Query Language.

EIRP: Equivalent Isotropic Radiated Power.

UHF: Ultra High Frequency.

RMSE: Root Mean Square Deviation.

UTM: Universal Traverse Mercator.

WGS84: World Geodetic System 84.